Introduction

In this module, we will learn how to use **functions** and **classes** to design software programs. Learning to create programs using functions and classes helps in structuring the code, simplifying the program's design, and controlling the solution's complexity.

Functions are the blocks in our program that encapsulate the most frequently used tasks and hide technical or unnecessary details. This type of encapsulation is called **abstraction**, and it is needed to build a maintainable program that can be easily modified, corrected, enhanced, and protected.

Classes are the templates or blueprints that can be instantiated to create **objects**. Those objects are only available during runtime. Objects interact and communicate to provide a solution and return output in an effective and maintainable manner.

The architecture of functions and classes concepts will be explored in this module, with a set of illustrative examples to clarify the usage of those concepts in designing a Python program.

Topics and Learning Objectives

Topics

- Using functions
- Introduction to objects and classes
- Inheritance and polymorphism

Learning Objectives

By the end of the module, you will be able to:

- Describe the importance of using functions.
- Construct and execute functions.
- Declare and instantiate class variables.
- Make use of object-oriented programming concepts, such as polymorphism and inheritance to achieve a maintainable, extensible, and reusable code.

Readings

Required Readings

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
    - Design with Functions, pages 167–182
    - Design with Classes, pages 293–308

Functions

Please watch this video presentation as it highlights the process of declaring a function in Python and describing its structure. Also, it explores three abstraction mechanisms that can be achieved by designing Python programs using functions. Those mechanisms are **elementary redundancy**, **systematic variations**, and **hiding complexity**. Examples and applications will be provided to articulate those concepts.

Ryecast Video:
CIND830 - Module 8: Functions

Download

Download the PDF file for **CIND830 - Module 8: Functions and Classes.**

Please note this PDF file will be used in the two videos of this module. In the PDF, you can use the top navigation bar to navigate between the topics covered in each video.

Download the **CIND830 - Module 8: Functions transcript**.

Classes

Please watch this video presentation as it demonstrates a different type of abstraction and shows how to design and implement classes in Python. The difference between classes and objects in Python is explored. Then, the purpose of using docstrings when creating a class will be highlighted. Also, the difference between mutator and accessor methods will be provided to learn which type of methods can be used in which situation and with which data structure. Finally, polymorphism and inheritance concepts will be explored as two of the most popular concepts in Python as an object-oriented programming language.

Ryecast Video:
CIND830 - Module 8: Classes

Download

Download the PDF file for **CIND830 - Module 8: Functions and Classes.** WE: please link green highlighted text to **CIND830 Module 8 Functions and Classes.pdf** and click on the topic "**Classes**" in the top navigation bar to take you to the beginning of this topic.

Download the **CIND830 - Module 8: Classes transcript.**

Self-Assessment Quiz

Please complete the following review activity. The activity is not graded.

Test Yourself

H5P Object:

Self-Assessment

Summary

In this module, we learned how to design Python programs using functions. We observed that there is a difference between declaring a function and calling or invoking it. A function can be invoked with writing its name, followed by a set of variables. Those variables will be passed to the function to place the pre-defined parameters in respective order. In any effective program, the resulting code can be extremely complex, difficult to verify, and almost impossible to maintain. In that context, we learned through this module three different mechanisms that can be applied to abstract our design. Since abstraction is useful to free the developer from the need to be concerned with additional details, we extended our knowledge to understand how to design programs using classes, and we learned that the class is the template from which objects can be instantiated. We ended our module by covering two of the most important concepts in object-oriented environments, being inheritance and polymorphism.

Activities

Discussion

Module 8 Discussion

This week, let's discuss functions and classes in the **Module 8 Discussion** topic in D2L under the **Communication** drop-down menu, then **Discussions**. Feel free to start a new thread or respond to someone else's post.

Lab Activity

Lab 6

Before starting Lab 6, be sure you have reviewed all the content in Module 8. The lab is a practical extension to apply your knowledge gained from this module. Download Lab 6 now and complete it by the end of this week. Your instructor will give you time to work on it before releasing the solutions in a folder called **Lab Solutions** in D2L under the **Course Materials** drop-down menu for you to verify your answers. The lab activity is not graded.

While working on your lab, post any questions, ideas, and strategies in the **Module 8 Discussion** topic in D2L. Feel free to read and respond to your peers' posts as well. You may also have the option of reaching out to your instructor during his/her virtual office hours.

Please note: The PDF for the lab activity may not be fully accessible for students who use assistive technology. If you require an accessible version, please contact your instructor.

Assessments

Quiz Module 8

Please go to your D2L course shell, under the **Assessment** drop-down menu, select **Quizzes**. Then click on the **Quiz Module 8 link** and follow the instructions to complete this quiz. The date and length of time have been set for

this quiz. **This quiz is a cumulative 10% of your final grade.**

Reminder

This week you are starting **Assignment 2 (worth 15% of your final grade)**. In D2L, go to the **Assessment** drop-down menu, then **Assignments**. Click on the **Assignment 2 link**. Read the description and follow the instructions to complete your assignment.

You are required to submit both the notebook file (in IPYNB format) and the exported version (either in PDF or HTML format) in the same Assignment 2 link in D2L. Failing to submit both files will be subject to a mark deduction. **Assignment 2 is due by the end of Module 12.**

References

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
- Python Software Foundation. (2020). Python 3.7.8 documentation. Retrieved from https://docs.python.org/3.7/index.html.